	L #	Hits	Search Text	DBs
1	L2	869	(access$3 writ$3 updat$3) near10 ((branch near5 (table history predict$3)) bht)	USPAT; US-PGPUB
2	L4	207	(access$3 writ$3 updat$3) near10 ((branch near5 (table history predict$3)) bht)	EPO; JPO; DERWENT; IBM_TDB
3	L7	220	2 near10 (control controlled controlling simultaneous$3 clock cycle)	USPAT; US-PGPUB
4	L6	29	4 near10 (control controlled controlling simultaneous$3 clock cycle)	EPO; JPO; DERWENT; IBM_TDB

F	7	-	1	0	0
E	6	-	1	0	0
D	3	5	1	1	0
C	O	2	1	1	0
B	O	-	1	0	1
A	O	-	1	0	1

Pointer ——→ (F row)

448
446
444

450
Return Stack
Storage 252
442
440

FIG. 4D

G	11	-	1	0	0
F	7	8	1	1	0
E	6	9	1	1	0
D	3	5	1	1	0
C	O	2	1	1	0
B	O	10	1	1	1
A	O	-	1	0	1

Pointer ——→ (G row)

448
446
444

452
450
Return Stack
Storage 252
442
440

FIG. 4E

	Document ID	U	Title	Current OR
1	JP 2001100995 A	☐	DEVICE AND METHOD FOR WRITE CONTROL OVER BRANCH HISTORY INFORMATION	
2	JP 11110224 A	☒	LANGUAGE PROCESSING DEVICE, ITS METHOD AND INFORMATION STORAGE MEDIUM	
3	JP 10171652 A	☒	METHOD AND DEVICE FOR UPDATING OF BRANCH HISTORY TABLE	
4	JP 09083552 A	☒	MULTIPROTOCOL CONTROL METHOD	
5	JP 06236270 A	☒	METHOD AND DEVICE FOR IMPROVING BRANCH HISTORY PREDICTING PRECISION IN SUPERSCALAR PROCESSOR SYSTEM	
6	JP 04242431 A	☒	INFORMATION PROCESSOR	
7	JP 02031237 A	☒	GENERATING SYSTEM FOR OPTIMUM BRANCH INSTRUCTION	
8	JP 01216427 A	☒	DATA PROCESSOR	
9	JP 63189943 A	☒	BRANCH FORECASTING CONTROL SYSTEM	
10	JP 63005442 A	☒	PROGRAM LOOP DETECTING AND STORING DEVICE	
11	JP 62262143 A	☒	GRAPHICAL LANGUAGE TRANSLATING DEVICE	
12	JP 62130460 A	☒	DOCUMENT GENERATION SYSTEM INSTRUCTED BY BRANCH OFFICE	
13	JP 60175148 A	☒	INSTRUCTION PREFETCHING DEVICE	
14	JP 60164842 A	☒	INSTRUCTION PREFETCHING DEVICE	
15	JP 57147197 A	☒	MEMORY PROTECTION SYSTEM	
16	EP 605876 A1	☒	Method and system for enhanced branch history prediction accuracy in a superscalar processor system.	
17	NA9001369	☒	Branch Resolution in the Cache	
18	NN83102283	☒	Functional AC Card Test Using a DC Tester	
19	US 6347369 B	☒	Microprocessor operating method involves reading counter values from branch history table in response to accessing branch history table using generated primary and secondary branch history table indexes	
20	US 20010020267 A	☒	Pipeline processing apparatus controls supply of updated latest branch prediction status to succeeding conditional branch instruction by comparing the branch instruction at preceding pipeline stage with succeeding stage	
21	US 5978907 A	☒	Updating method of storage arrays in superscalar microprocessor	
22	US 5878255 A	☒	Delayed update unit for branch prediction array for superscalar microprocessor	

Tag. Rename buffer 14 therefore indicates a "hit" to forwarding circuit 18. Forwarding circuit 18 selects the operand present on operand bus/RB 28 to forward to execution unit 18 via operand bus 20. Again, rename buffer 14 forwards the Data field having an Architectural Register File Tag field matching the requested Architectural Register File Tag, a high Valid bit and a high MRA bit for forwarding circuit 18 via operand bus/RB 28. In this scenario, however, the entry is assumed to be "not present." The Data Present bit associated with the selected memory element of rename buffer 14 therefore contains a zero logic state. In the case of an invalid data signal (data not present), execution unit 12 ignores the forwarded operand but latches the rename tag forwarded to it by execution unit 12.

During a subsequent clock cycle, execution unit 12 will monitor result bus 24 and result/request tag bus 26. This process is called "snooping." A second execution unit will eventually return a result that is the desired operand with the operand's associated rename tag. The second execution unit will forward the operand and rename tag to rename buffer 14 via result bus 24 and result/request tag bus 26, respectively. Execution unit 12 will be able to identify this operand by the rename tag on result/request tag bus 26 and can latch the operand at the same time rename buffer 14 latches the operand and rename tag.

FIGS. 3 through 7 depict the contents of the rename buffer 14 at various sequential times. In the illustrated examples, rename buffer 14 has seven memory elements, labeled A through G, that store eight results in eight Data fields for eight architectural registers, labeled 0 through 7 to avoid confusion. In practice, the seven memory elements and eight architectural registers would be identified with binary numbers. Blank entries in FIGS. 3 through 7 indicate that the value of the field is not relevant to the discussion.

Normal Operation

FIG. 3 depicts the contents of rename buffer 14 illustrated in FIG. 2 after memory elements A, B and C have been allocated to three instructions. The Data field of memory element A already contains the result of an instruction (Data Present bit set to a one logic state) that will write to architectural register #4. The Data fields of memory elements B and C will contain the results of two instructions that will write to architectural registers #5 and #7, respectively. These latter instructions have not completed as indicated by the zero logic state of the two corresponding Data Present bits. The MRA bit of all the memory elements reflect that each is or will be the most recent version of some architectural register, here architectural registers #4, #5 and #7. Memory elements D through G are not valid as indicated by the zero logic state of each memory element's Valid bit.

FIG. 4 depicts the contents of rename buffer 14 illustrated in FIG. 2 after an additional memory element, memory element D, has been allocated to some instruction. This additional instruction will write to architectural register #4 like the instruction associated with memory element A. At instruction issue, the MRA bit of memory element D is set to a one logic state and the MRA bit of memory element A is set to a zero logic state. As a result, any later instruction that requests the contents of architectural register #4 will receive the Data field or the Rename Tag field of memory element D depending upon the Data Present bit of memory element D. At the depicted moment, the result of the instruction associated with memory element D is not complete. In that case, rename buffer 14 will forward the Rename Tag field to any execution unit requesting architectural register #4.

Speculative Branching

Rename buffer 14 may be modified to operate within a data processor that supports speculative branching. A data processor that implements a speculative branching scheme is one that branches to one of two (or more) possible instructions before the data processor actually determines whether the selected instruction stream is in fact the correct instruction stream. The data processor will continue along its new instruction stream if it later determines that the new instruction stream is correct. The data processor will stop issuing instructions along the selected path, however, if it determines that it "guessed" incorrectly. The data processor will then attempt to reverse the affects of the instructions issued along the incorrect path and issue instructions along the correct path. If it should be understood that either one of the possible instructions streams may serially follow the first instruction stream or may require a jump to a different point in software. The speculative branching feature is illustrated below in FIGS. 5 through 7.

FIG. 5 depicts the contents of rename buffer 14 illustrated in FIG. 2 if data processor 10 takes a speculative branch after issuing the four instructions described above in connection with FIGS. 3 and 4. Rename buffer 14 saves a copy of each MRA bit in the Shadow MRA bit associated with each MRA bit. Specifically, the Shadow MRA bit of A memory element equals the MRA bit of A memory element, the Shadow MRA bit of B memory element equals the MRA bit of B memory element, etc.

FIG. 6 depicts the contents of rename buffer 14 illustrated in FIG. 2 after data processor 10 issues one instruction from the new instruction stream. This next instruction is allocated the memory element E and will write to architectural register #5. As a result, rename buffer 14 sets the MRA bit of memory element E to a zero logic state to reflect that memory element B to a zero logic state to reflect that memory element E will contain the most recently allocated version of architectural register #5. The Shadow MRA bits are not modified.

If data processor 10 "guessed" the correct instruction path when it branched speculatively, then data processor 10 will continue along the same instruction stream. Rename buffer 14 will ignore its Shadow MRA bits until data processor 10 takes a second speculative branch. At that point, rename buffer 14 will again copy the MRA bits into the Shadow MRA bits, overwriting the vector illustrated in FIG. 6. If, however, data processor 10 guessed incorrectly, then rename buffer 14 will copy the Shadow bits back into the associated MRA bits. Rename buffer 14 or, perhaps, a branch unit of data processor 10 will also invalidate all instructions along the incorrect instruction stream.

FIG. 7 depicts the contents of rename buffer 14 illustrated in FIG. 2 after data processor 10 determines that it followed an incorrect instruction stream. Rename buffer 14 invalidates memory element E and copies the contents of the Shadow bits back into the MRA bits. In the present illustration only one instruction issued before data processor 10 determined the correct instruction stream to follow. There-fore, only one memory element needed to be invalidated and only one MRA bit needed to be restored. The disclosed procedure, however, will produce the same results given any number of incorrectly issued instructions.

The number of Shadow bits per memory element may be increased to support multiple levels of speculative branch-ing. For instance, it may be desirable for data processor 10 to be able to take a second speculative branch before it determines if the first speculative branch was correct. In this case, a second Shadow MRA bit may be provided for each

	Document ID	U	Title	Current OR
23	US 5875324 A	☒	Delayed updating mechanism for branch prediction array in superscalar microprocessor	
24	JP 09244892 A	☒	CPU for computer - includes effective request flag in read-out buffer which indicates effect of error in branch prediction on read-out data field	
25	AU 9523231 A	☒	Memory control system for private branch exchange - has memory table in CPU with read circuit accessing information file in secondary memory for storage in table for access unit control	
26	EP 667618 A	☒	Access control method for particular address on disc - providing branch data for one control method in table used by different control method	
27	EP 651331 A	☐	Write buffer for super-pipelined super-scalar microprocessor - directs each write to memory to write buffer rather than memory bus or cache memory and writes contents of write buffer to cache or main memory when memory bus or cache becomes available	
28	EP 605876 A	☐	Branch history prediction for processor - using branch history table with predictive field for each fetch cycle in multiple instruction access accessed by partial instruction address for updating	
29	EP 258453 B	☐	Instruction prefetch control appts. for data processor - has branch history table storing paired branch instruction address and branch destination address of branch instruction	

memory element. Upon execution of the second outstanding speculative branch, the contents of the MRA bits would be stored in the second Shadow MRA bits. Rename buffer 14 will then restore the Nth Shadow MRA bits to the MRA bits if the data processor's branch until determines that the Nth outstanding speculative branch is incorrect (where N is an integer index). In general, one Shadow MRA bit may be provided for each level of speculative branching that is desired.

FIG. 8 depicts a flow chart 36 of one set of steps operable to implement the disclosed invention. Rename buffer 14 performs flow chart 36 each time an operand is requested by some execution unit. Rename buffer 14 compares the Archi-tectural Register File Tag of the requested operand to each Architectural Register File Tag field in rename buffer 14 for a match, step 38. Rename buffer 14 then branches to a path 40 or to a path 42 depending upon whether rename buffer 14 finds a match or does not find a match, respectively. Step 38 is more fully described below in connection with FIG. 9.

Continuing along path 40, rename buffer 14 will transmit a "hit" signal to forwarding circuit 40 indicating that for-warding circuit 18 should forward the operand supplied by rename buffer 14 to operand bus 20, step 44. Rename buffer 14 will then forward the matching rename tag field and Data field (if any) and the Data Present bit to execution unit 18, step 46. As described above, execution unit 18 will disregard certain of these fields depending upon the value of the Data Present field.

Continuing along path 42, rename buffer 14 will transmit a "miss" signal to forwarding circuit 18 indicating that forwarding circuit 18 should forward the operand supplied by architectural register file 16, step 48. Rename buffer 14 may or may not forward any other data to execution unit 12 depending upon the implementation details of rename buffer 14. Regardless, according to the protocol described above, any data forwarded to execution unit 12 will be ignored given the valid data signal supplied by architectural register file 16.

Both paths within flow chart 36 then merge at the end of flow chart 36.

FIG. 9 depicts a flow chart of one set of steps operable to implement step 38 illustrated in FIG. 8. Rename buffer 14 performs a tag compare, step 50. As described above, rename buffer compares, respectively, (1) the tag of the requested operand, a one logic level, and a one logic level to (2) the Architectural Register File Tag field, the Valid bit, and the MRA bit of each memory element 35. Rename buffer 14 then branches, in step 52, to path 40 or 42 depending upon whether or not it is found a match in step 50. The details of compare step 50 will depend upon the hardware implementation of the memory elements in rename buffer 14. For instance, if rename buffer 14 is designed as a small block of Random Access Memory ("RAM"), then step 50 may involve an iterative loop in which a memory cell is addressed according to an indexed counter, the memory cell contents are compared, and the counter is incremented if no match is found. Conversely, rename buffer 14 may be a small block of content address-able memory ("CAM") cells. If the rename buffer 14 is a block of CAM cells, then the contents of each CAM cell may be compared to the input operand tag in parallel with each other. The CAM cell having the desired fields will automati-cally output its Rename Tag field, DATA field, and Data Present bit.

FIG. 10 depicts a flow chart 54 of one set of steps operable to implement the disclosed invention. In particular, flow

chart 54 illustrates how rename buffer 14 may incorporate the speculative branching feature described above in con-nection with FIGS. 5 through 7. Rename buffer 14 may execute the steps depicted in FIG. 10 at the beginning of each data processor clock cycle. Rename buffer 14 deter-mines if data processor 10 takes a speculative branch, step 56. If data processor 10 has taken a speculative branch, then rename buffer 14 copies each MRA bit to the associated Shadow MRA bit, then flows to step 58. Flow chart 54 then flows to step 60. If data processor 10 does not take a speculative branch, then rename buffer 14 flows directly to step 60 from step 58.

Rename buffer 14 then determines if a data processor indicates that data processor 10 followed an incorrect instruction stream, step 60. If data processor 10 did follow an incorrect instruction stream, then rename buffer 14 invali-dates each Valid bit associated with an instruction that followed the speculative branch instruction, step 62. Rename buffer 14 also copies each Shadow MRA bit back to the corresponding MRA bit, step 64. If data processor 10 then flows to the end of flow chart 54. If data processor 10 did not follow an incorrect instruction stream, then rename buffer 14 flows directly to the end of flow chart 54 from step 60.

Although the present invention has been described with reference to a specific embodiment, further modifications and improvements will occur to those skilled in the art. For instance, many of the functions attributed above to a par-ticular portion of data processor 10 may be performed by a different portion of a particular data processor. The name of the unit performing a necessary function or the division of labor among a group of units, is therefore, not part of the invention. It is to be understood therefore, that the invention encompasses all such modifications that do not depart from the spirit and scope of the invention as defined in the appended claims.

What is claimed is:
1. A data processor comprising:
a plurality of execution units, the execution units operable to perform a plurality of instructions received from a memory system, at least one of the plurality of instruc-tions requiring an operand, at least one of the plurality of instructions generating a result;
a plurality of architectural registers coupled to at least one of the execution units, the plurality of architectural registers supplying the at least one of the plurality of architec-tural registers periodically receiving the result of the at least one instruction generating a result;
a plurality of memory means, each memory means storing a result, a tag representative of the architectural register associated with the result and a first-most-recently-allocated bit and a second-most-recently-allocated bit each associated with the result;
allocation means coupled to at least one of the execution units and to the plurality of memory means, the allo-cation means periodically receiving a first tag identi-fying an architectural register, storing the first tag in a predetermined one of the memory means, setting the first-most-recently-allocated bit associated with the predetermined one of memory means to a first logic state, setting each first-most-recently-allocated bit of a subset of the memory elements to a second logic state, the subset of memory elements comprising a tag logi-cally equivalent to the first tag, and storing each first-most-recently-allocated bit in an associated second-most-recently-allocated bit location responsive to a speculative branch;

	Document ID	U	Title	Current OR
1	US 2002017432 8 A1	☐	Method for cancelling speculative conditional delay slot instructions	712/235
2	US 2002017432 2 A1	☒	Method for cancelling conditional delay slot instructions	712/218
3	US 2002008779 4 A1	☒	Apparatus and method for speculative prefetching after data cache misses	711/126
4	US 2002008331 0 A1	☒	METHOD AND APPARATUS FOR PREDICTING LOOP EXIT BRANCHES	712/233
5	US 2002008067 7 A1	☒	Semiconductor memory device	365/233
6	US 2002007833 2 A1	☒	Conflict free parallel read access to a bank interleaved branch predictor in a processor	712/240
7	US 2002006937 5 A1	☒	System, method, and article of manufacture for data transfer across clock domains	713/400
8	US 2001002751 5 A1	☒	Apparatus and method of controlling instruction fetch	712/207
9	US 2001002197 4 A1	☒	Branch predictor suitable for multi-processing microprocessor	712/240
10	US 2001002026 7 A1	☒	Pipeline processing apparatus with improved efficiency of branch prediction, and method therefor	712/239
11	US 2001002026 5 A1	☒	Data processor with multi-command instruction words	712/24
12	US 2001001690 3 A1	☒	Software branch prediction filtering for a microprocessor	712/239
13	US 2001001134 6 A1	☒	Branch prediction method, arithmetic and logic unit, and information processing apparatus	712/239
14	US 2001000753 3 A1	☒	Non-volatile semiconductor memory device and semiconductor disk device	365/185 .11
15	US 6467064 B1	☒	Viterbi decoder	714/795
16	US 6453278 B1	☒	Flexible implementation of a system management mode (SMM) in a processor	703/27
17	US 6445615 B2	☒	Non-volatile semiconductor memory device and semiconductor disk device	365/185 .11
18	US 6442679 B1	☒	Apparatus and method for guard outcome prediction	712/218

11

forwarding means coupled to at least one of the execution units and to the plurality of memory means, the forwarding means receiving a request for an operand, the request comprising the first tag, the forwarding means comparing the first tag to each tag stored in the plurality of memory means, and forwarding a selected result to an execution unit responsive to the step of comparing, the selected result associated with a selected one of the plurality of memory means, the selected one of the plurality of memory means comprising a tag and a first-most-recently-allocated bit corresponding in logic state to the first one of the architectural registers and to the first logic state of the first-most-recently-allocated bit, respectively; and

2. A method of operating a data processor comprising the steps of:

at a first time, storing a first tag and a first first-most-recently-allocated bit in a tag field and in a first-most-recently-allocated-bit-field, respectively, of a first one of a plurality of sets, each one of the plurality of sets comprising a tag field, a first-most-recently-allocated-bit-field, and a second first-most-recently-allocated-bit-field, the first tag identifying an address of a first memory cell associated with a first result being an output of a first instruction executed by an execution unit of the data processor, the first first-most-recently-allocated bit associated with the first result, assuming a first logic state;

12

at a second time subsequent to the first time, storing a second tag and a second first-most-recently-allocated bit in a second one of the plurality of sets, the second tag identifying an address of a memory cell associated with a second result and the second first-most-recently-allocated bit associated with the second result, being an input of a second instruction executed by the second execution unit, the second first-most-recently-allocated bit assuming the first logic state, and storing a second logic state in the first-most-recently-allocated bit field of a subset of the plurality of sets, each of the contents of the tag fields of the subset of the plurality of sets logically equivalent to the second tag;

at a third time, receiving a requested tag of a requested result, comparing the requested tag and the first logic result, comparing the requested tag and the first logic state of the tag field and the first-most-recently-allocated bit field, respectively, of each of the plurality of sets, forwarding a result associated with a selected set, the contents of the tag field and the contents of the first-most-recently-allocated-bit-field of the selected set logically equivalent to the requested tag and the first set logically equivalent to the requested tag and the first logic state;

at a fourth time subsequent to the first time, receiving the first result from a communication bus within the data processor and storing the first result in the first memory cell; and

at a fifth time subsequent to the first time, issuing a speculative branch instruction to the execution unit and storing the contents of each first-most-recently-allocated-most-recently-allocated bit field.

* * * * *

	Document ID	U	Title	Current OR
19	US 64386 82 B1	☒	Method and apparatus for predicting loop exit branches	712/241
20	US 64386 64 B1	☒	Microcode patch device and method for patching microcode using match registers and patch routines	711/154
21	US 64271 92 B1	☒	Method and apparatus for caching victimized branch predictions	711/133
22	US 64250 75 B1	☒	Branch prediction device with two levels of branch prediction cache	712/239
23	US 63973 26 B1	☒	Method and circuit for preloading prediction circuits in microprocessors	712/240
24	US 63935 49 B1	☒	Instruction alignment unit for routing variable byte-length instructions	712/204
25	US 63743 51 B1	☒	Software branch prediction filtering for a microprocessor	712/239
26	US 63599 09 B1	☒	Switch device for relayin G cells or packets on demand	370/522
27	US 63538 82 B1	☒	Reducing branch prediction interference of opposite well behaved branches sharing history entry by static prediction correctness based updating	712/239
28	US 63517 97 B1	☒	Translation look-aside buffer for storing region configuration bits and method of operation	711/207
29	US 63517 89 B1	☒	Built-in self-test circuit and method for validating an associative data array	711/128
30	US 63473 69 B1	☒	Method and circuit for single cycle multiple branch history table access	712/240
31	US 63413 48 B1	☒	Software branch prediction filtering for a microprocessor	712/239
32	US 63361 78 B1	☒	RISC86 instruction set	712/23
33	US 63321 91 B1	☒	System for canceling speculatively fetched instructions following a branch mis-prediction in a microprocessor	712/240
34	US 63306 62 B1	☒	Apparatus including a fetch unit to include branch history information to increase performance of multi-cylce pipelined branch prediction structures	712/236
35	US 63144 93 B1	☒	Branch history cache	711/137
36	US 63016 47 B1	☒	Real mode translation look-aside buffer and method of operation	711/207
37	US 62826 29 B1	☒	Pipelined processor for performing parallel instruction recording and register assigning	712/23
38	US 62791 07 B1	☒	Branch selectors associated with byte ranges within an instruction cache for rapidly identifying branch predictions	712/239
39	US 62726 24 B1	☒	Method and apparatus for predicting multiple conditional branches	712/239
40	US 62694 36 B1	☒	Superscalar microprocessor configured to predict return addresses from a return stack storage	712/23
41	US 62667 52 B1	☒	Reverse TLB for providing branch target address in a microprocessor having a physically-tagged cache	711/200

United States Patent [19]

Tran et al.

[11]	Patent Number: **5,881,278**
[45]	Date of Patent: **Mar. 9, 1999**

[54] RETURN ADDRESS PREDICTION SYSTEM WHICH ADJUSTS THE CONTENTS OF RETURN STACK STORAGE TO ENABLE CONTINUED PREDICTION AFTER A MISPREDICTED BRANCH

[75] Inventors: **Thang M. Tran; Rupaka Mahalingaiah**, both of Austin, Tex.

[73] Assignee: **Advanced Micro Devices, Inc.,** Sunnyvale, Calif.

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,044,338	8/1977	Wolf .
4,453,212	6/1984	Gaither et al. .

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0259095	3/1988	European Pat. Off. .
0381471	8/1990	European Pat. Off. .
0459232	12/1991	European Pat. Off. .
0649084	4/1995	European Pat. Off. .
2263985	8/1993	United Kingdom .
2263987	8/1993	United Kingdom .
2281422	3/1995	United Kingdom .
2287111	9/1995	United Kingdom .

OTHER PUBLICATIONS

Intel, "Chapter 2: Microprocessor Architecture Overview," pp. 2–1 through 2–4.

Michael Slater, "AMD's K5 Designed to Outrun Pentium," Microprocessor Report, vol. 8, No. 14, Oct. 24, 1994, 7 pages.

Sebastian Rupley and John Clyman, "P6: The Next Step?," PC Magazine, Sep. 12, 1995, 16 pages.

Tom R. Halfhill, "AMD K6 Takes on Intel P6,"BYTE, Jan. 1996, 4 pages.

Dialog, Microprocessor Report, Oct. 24, 1994, pp. 1–7.

Foreign Search Report dated Feb. 26, 1997 for PCT/US96/11842.

Tomasula, R. M., "An Efficient Algorithm for Exploiting Multiple Arithmetic Units," IBM Journal, Jan. 1967, pp. 25–33.

Dialog, Microprocessor Report, Oct. 24, 1994, pp. 1–7.

IBM Technical Disclosure Bulletin, "Subroutine Call/Return Stack," Apr. 30, 1988, vol. 30, No. 11, pp. 221–225.

IBM Technical Disclosure Bulletin, "Highly Accurate Subroutine Stack Prediction Mechnanism," Mar. 10, 1986, vol. 28, No. 10, pp. 4635–4637.

Yeh and Patt, "Branch History Table Indexing to Prevent Pipeline Bubbles in Wide–Issue Superscalar Processors," Dept. of Electrical Engineering, University of Michigan, An Arbor, 1993, pp. 164–175.

[57] **ABSTRACT**

A return prediction unit is provided which is configured to predict return addresses for return instructions according to a return stack storage included therein. The return stack storage is a stack structure configured to store return addresses associated with previously detected call instructions. Return addresses may be predicted for return instructions early in the instruction processing pipeline of the microprocessor. In one embodiment, the return stack storage additionally stores a call tag and a return tag with each return address. The call tag and return tag respectively identify call and return instructions associated with the return address. These tags may be compared to a branch tag conveyed to the return prediction unit upon detection of a branch misprediction. The results of the comparisons may be used to adjust the contents of the return stack storage with respect to the misprediction. The return prediction unit may continue to predict return addresses correctly following a mispredicted branch instruction.

39 Claims, 45 Drawing Sheets
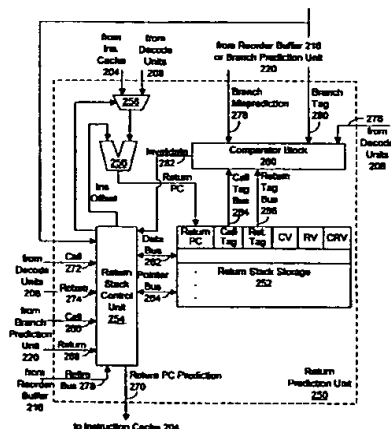
	Document ID	U	Title	Current OR
42	US 62533 16 B1	☒	Three state branch history using one bit in a branch prediction mechanism	712/239
43	US 62471 23 B1	☒	Branch prediction mechanism employing branch selectors to select a branch prediction	712/239
44	US 62471 22 B1	☒	Method and apparatus for performing branch prediction combining static and dynamic branch predictors	712/239
45	US 62232 80 B1	☒	Method and circuit for preloading prediction circuits in microprocessors	712/240
46	US 62021 42 B1	☒	Microcode scan unit for scanning microcode instructions using predecode data	712/204
47	US 61924 68 B1	☒	Apparatus and method for detecting microbranches early	712/231
48	US 61890 91 B1	☒	Apparatus and method for speculatively updating global history and restoring same on branch misprediction detection	712/240
49	US 61890 68 B1	☒	Superscalar microprocessor employing a data cache capable of performing store accesses in a single clock cycle	711/3
50	US 61856 74 B1	☒	Method and apparatus for reconstructing the address of the next instruction to be completed in a pipelined processor	712/230
51	US 61784 98 B1	☒	Storing predicted branch target address in different storage according to importance hint in branch prediction instruction	712/239
52	US 61675 10 A	☒	Instruction cache configured to provide instructions to a microprocessor having a clock cycle time less than a cache access time of said instruction cache	712/239
53	US 61516 72 A	☒	Methods and apparatus for reducing interference in a branch history table of a microprocessor	712/239
54	US 61417 48 A	☒	Branch selectors associated with byte ranges within an instruction cache for rapidly identifying branch predictions	712/239
55	US 61417 40 A	☒	Apparatus and method for microcode patching for generating a next address	711/215
56	US 61254 44 A	☒	Millimode capable computer system providing global branch history table disables and separate millicode disables which enable millicode disable to be turned off for some sections of code execution but not disabled for all	712/245
57	US 61192 22 A	☒	Combined branch prediction and cache prefetch in a microprocessor	712/238
58	US 61192 20 A	☒	Method of and apparatus for supplying multiple instruction strings whose addresses are discontinued by branch instructions	712/235
59	US 61087 76 A	☒	Globally or selectively disabling branch history table operations during sensitive portion of millicode routine in millimode supporting computer	712/240
60	US 61087 74 A	☒	Branch prediction with added selector bits to increase branch prediction capacity and flexibility with minimal added bits	712/240
61	US 61065 73 A	☒	Apparatus and method for tracing microprocessor instructions	717/128
62	US 61015 95 A	☒	Fetching instructions from an instruction cache using sequential way prediction	712/205
63	US 61015 77 A	☒	Pipelined instruction cache and branch prediction mechanism therefor	711/125

U.S. PATENT DOCUMENTS

4,504,927	3/1985	Callan	364/900
4,807,115	2/1989	Torng	
4,858,105	8/1989	Kuriyama et al.	
5,136,697	8/1992	Johnson	
5,179,673	1/1993	Steely, Jr. et al.	395/586
5,222,220	6/1993	Mehta	395/375
5,226,126	7/1993	McFarland et al.	
5,226,130	7/1993	Favor et al.	
5,274,817	12/1993	Stahl	395/700
5,313,634	5/1994	Eickemeyer	395/700
5,339,422	8/1994	Brender et al.	395/700
5,355,459	10/1994	Matsuo et al.	395/589
5,454,087	9/1995	Narita et al.	395/375
5,526,498	6/1996	Matsuo et al.	395/375
5,564,118	10/1996	Steely, Jr. et al.	395/375
5,574,871	11/1996	Hoyt et al.	395/376
5,584,001	12/1996	Hoyt et al.	395/585
5,604,877	2/1997	Hoyt et al.	395/590
5,606,682	2/1997	McGarity	395/414
5,623,614	4/1997	Van Dyke et al.	395/587
5,649,225	7/1997	White et al.	395/800
5,655,098	8/1997	Witt et al.	395/386

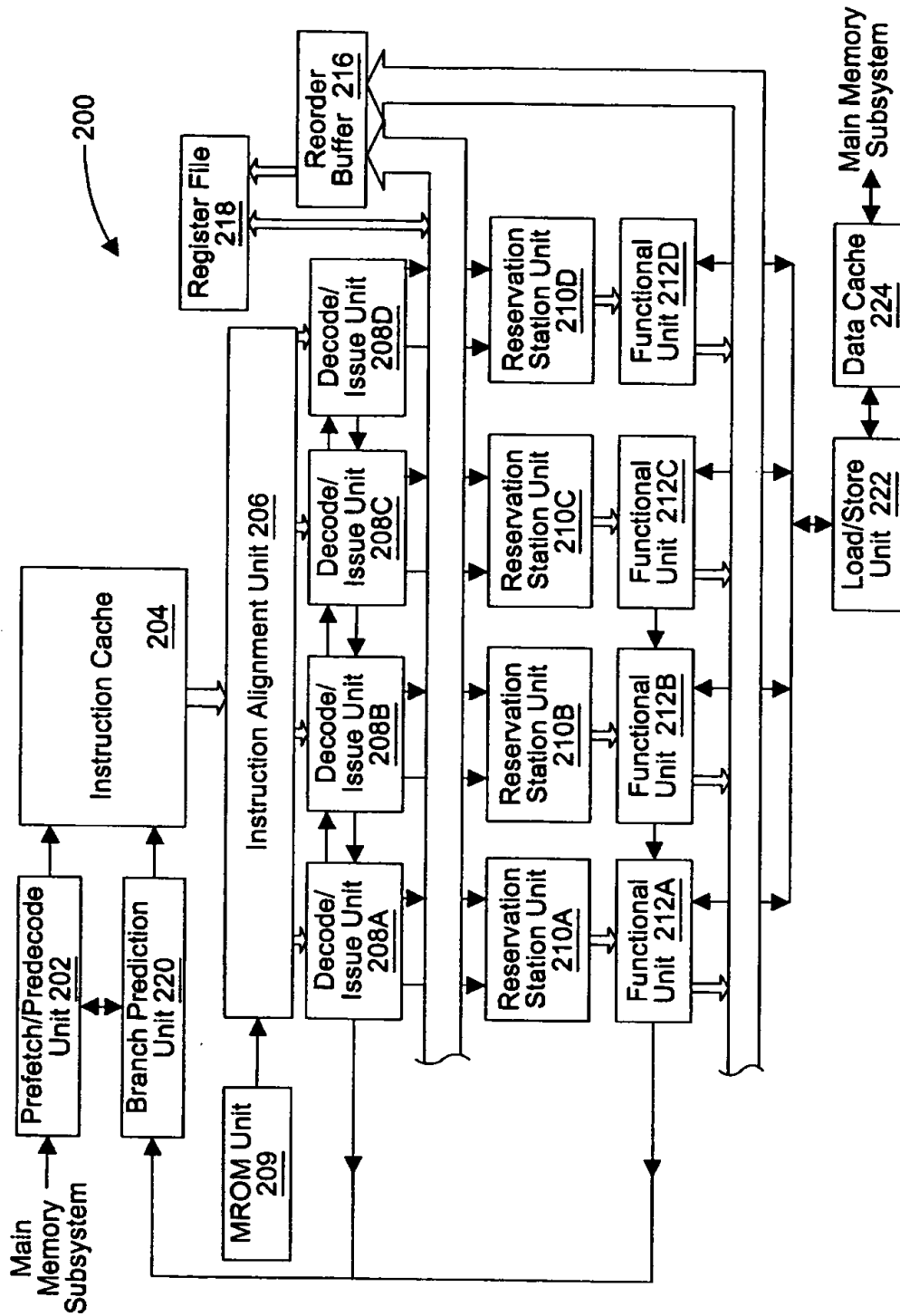|	Document ID	U	Title	Current OR
64	US 6093213 A	☒	Flexible implementation of a system management mode (SMM) in a processor	703/27
65	US 6081887 A	☒	System for passing an index value with each prediction in forward direction to enable truth predictor to associate truth value with particular branch instruction	712/239
66	US 6079006 A	☒	Stride-based data address prediction structure	711/213
67	US 6079005 A	☒	Microprocessor including virtual address branch prediction and current page register to provide page portion of virtual and physical fetch address	711/213
68	US 6079003 A	☒	Reverse TLB for providing branch target address in a microprocessor having a physically-tagged cache	711/200
69	US 6073230 A	☒	Instruction fetch unit configured to provide sequential way prediction for sequential instruction fetches	712/205
70	US 6065115 A	☒	Processor and method for speculatively executing instructions from multiple instruction streams indicated by a branch instruction	712/235
71	US 6065091 A	☒	Translation look-aside buffer slice circuit and method of operation	711/3
72	US 6055630 A	☒	System and method for processing a plurality of branch instructions by a plurality of storage devices and pipeline units	712/240
73	US 6052773 A	☒	DPGA-coupled microprocessors	712/43
74	US 6044478 A	☒	Cache with finely granular locked-down regions	714/42
75	US 6032252 A	☒	Apparatus and method for efficient loop control in a superscalar microprocessor	712/233
76	US 6032241 A	☒	Fast RAM for use in an address translation circuit and method of operation	711/207
77	US 6016545 A	☒	Reduced size storage apparatus for storing cache-line-related data in a high frequency microprocessor	712/238
78	US 6014741 A	☒	Apparatus and method for predicting an end of a microcode loop	712/233
79	US 6014734 A	☒	Superscalar microprocessor configured to predict return addresses from a return stack storage	712/23
80	US 6009513 A	☒	Apparatus and method for detecting microbranches early	712/231
81	US 6006324 A	☒	High performance superscalar alignment unit	712/204
82	US 6003128 A	☒	Number of pipeline stages and loop length related counter differential based end-loop prediction	712/241
83	US 5996071 A	☒	Detecting self-modifying code in a pipelined processor with branch processing by comparing latched store address to subsequent target address	712/238
84	US 5995749 A	☒	Branch prediction mechanism employing branch selectors to select a branch prediction	712/239
85	US 5987561 A	☒	Superscalar microprocessor employing a data cache capable of performing store accesses in a single clock cycle	711/3
86	US 5983337 A	☒	Apparatus and method for patching an instruction by providing a substitute instruction or instructions from an external memory responsive to detecting an opcode of the instruction	712/32

12/05/2002, EAST Version: 1.03.0002

200

Main Memory Subsystem

Prefetch/Predecode Unit 202

Branch Prediction Unit 220

MROM Unit 209

Instruction Cache 204

Instruction Alignment Unit 206

Decode/ Issue Unit 208A

Decode/ Issue Unit 208B

Decode/ Issue Unit 208C

Decode/ Issue Unit 208D

Register File 218

Reorder Buffer 216

Reservation Station Unit 210A

Reservation Station Unit 210B

Reservation Station Unit 210C

Reservation Station Unit 210D

Functional Unit 212A

Functional Unit 212B

Functional Unit 212C

Functional Unit 212D

Load/Store Unit 222

Data Cache 224

Main Memory Subsystem

FIG. 1

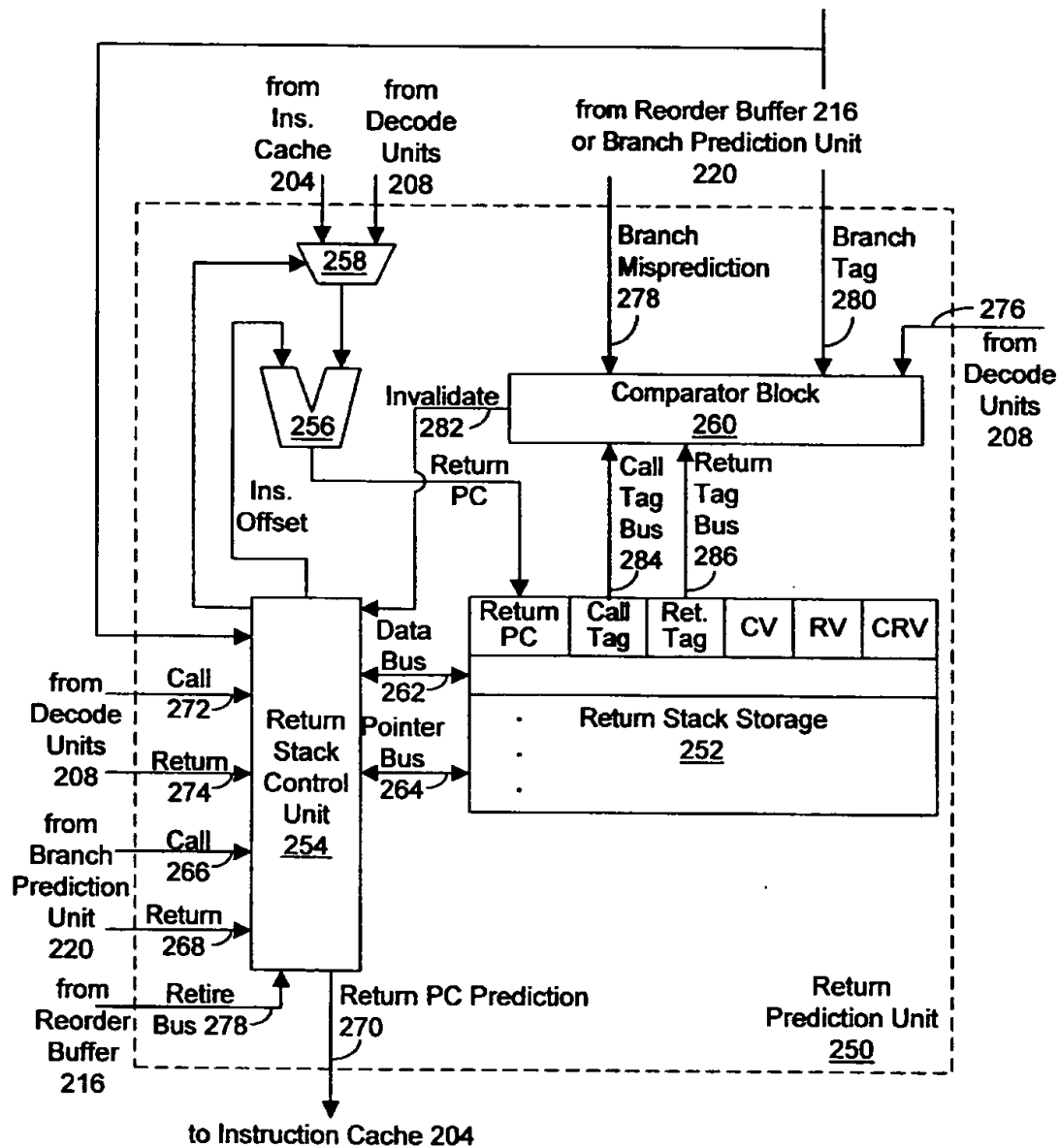|	Docum ent ID	U	Title	Current OR
87	US 59789 08 A	☒	Computer instruction supply	712/240
88	US 59789 07 A	☒	Delayed update register for an array	712/239
89	US 59789 06 A	☒	Branch selectors associated with byte ranges within an instruction cache for rapidly identifying branch predictions	712/239
90	US 59745 43 A	☒	Apparatus and method for performing subroutine call and return operations	712/240
91	US 59705 09 A	☒	Hit determination circuit for selecting a data set based on miss determinations in other data sets and method of operation	711/128
92	US 59681 69 A	☒	Superscalar microprocessor stack structure for judging validity of predicted subroutine return addresses	712/239
93	US 59681 63 A	☒	Microcode scan unit for scanning microcode instructions using predecode data	712/204
94	US 59616 38 A	☒	Branch prediction mechanism employing branch selectors to select a branch prediction	712/239
95	US 59548 16 A	☒	Branch selector prediction	712/239
96	US 59499 95 A	☒	Programmable branch prediction system and method for inserting prediction operation which is independent of execution of program code	712/239
97	US 59467 18 A	☒	Shadow translation look-aside buffer and method of operation	711/207
98	US 59467 05 A	☒	Avoidance of cache synonyms	711/108
99	US 59408 58 A	☒	Cache circuit with programmable sizing and method of operation	711/139
100	US 59387 61 A	☒	Method and apparatus for branch target prediction	712/238
101	US 59352 39 A	☒	Parallel mask decoder and method for generating said mask	712/224
102	US 59336 29 A	☒	Apparatus and method for detecting microbranches early	712/248
103	US 59336 26 A	☒	Apparatus and method for tracing microprocessor instructions	712/227
104	US 59336 18 A	☒	Speculative register storage for storing speculative results corresponding to register updated by a plurality of concurrently recorded instruction	712/217
105	US 59283 58 A	☒	Information processing apparatus which accurately predicts whether a branch is taken for a conditional branch instruction, using small-scale hardware	712/239
106	US 59266 42 A	☒	RISC86 instruction set	712/1
107	US 59266 34 A	☒	Limited run branch prediction	712/239
108	US 59207 13 A	☒	Instruction decoder including two-way emulation code branching	712/236
109	US 59095 67 A	☒	Apparatus and method for native mode processing in a RISC-based CISC processor	712/208

FIG. 2

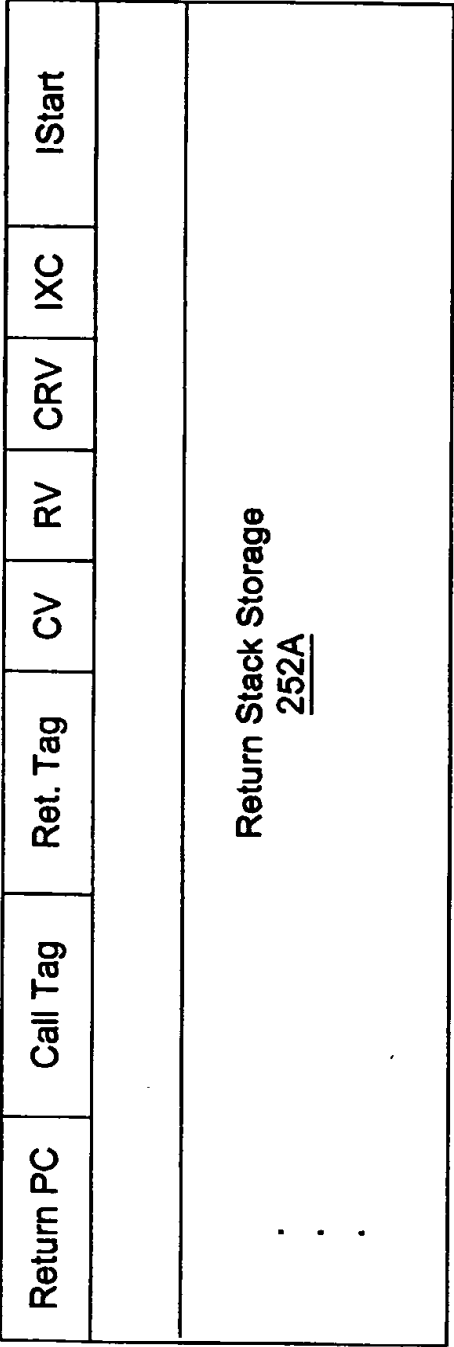|	Document ID	U	Title	Current OR
110	US 5907860 A	☒	System and method of retiring store data from a write buffer	711/117
111	US 5901307 A	☒	Processor having a selectively configurable branch prediction unit that can access a branch prediction utilizing bits derived from a plurality of sources	712/240
112	US 5898865 A	☒	Apparatus and method for predicting an end of loop for string instructions	712/239
113	US 5898286 A	☒	Digital servo control system for a data recording disk file with improved saturation modelling	318/569
114	US 5892936 A	☒	Speculative register file for storing speculative register states and removing dependencies between instructions utilizing the register	712/216
115	US 5887152 A	☒	Load/store unit with multiple oldest outstanding instruction pointers for completing store and load/store miss instructions	712/217
116	US 5881278 A	☒	Return address prediction system which adjusts the contents of return stack storage to enable continued prediction after a mispredicted branch	712/242
117	US 5878255 A	☒	Update unit for providing a delayed update to a branch prediction array	712/240
118	US 5875325 A	☒	Processor having reduced branch history table size through global branch history compression and method of branch prediction utilizing compressed global branch history	712/240
119	US 5875324 A	☒	Superscalar microprocessor which delays update of branch prediction information in response to branch misprediction until a subsequent idle clock	712/238
120	US 5875315 A	☒	Parallel and scalable instruction scanning unit	712/204
121	US 5873115 A	☒	Cache memory	711/129
122	US 5872989 A	☒	Processor having a register configuration suited for parallel execution control of loop processing·	712/23
123	US 5867724 A	☒	Integrated routing and shifting circuit and method of operation	712/22
124	US 5867698 A	☒	Apparatus and method for accessing a branch target buffer	712/238
125	US 5864707 A	☒	Superscalar microprocessor configured to predict return addresses from a return stack storage	712/23
126	US 5864697 A	☒	Microprocessor using combined actual and speculative branch history prediction	712/240
127	US 5860104 A	☒	Data cache which speculatively updates a predicted data cache storage location with store data and subsequently corrects mispredicted updates	711/137
128	US 5860017 A	☒	Processor and method for speculatively executing instructions from multiple instruction streams indicated by a branch instruction	712/23
129	US 5859991 A	☒	Parallel and scalable method for identifying valid instructions and a superscalar microprocessor including an instruction scanning unit employing the method	712/204
130	US 5854921 A	☒	Stride-based data address prediction structure	712/239
131	US 5854913 A	☒	Microprocessor with an architecture mode control capable of supporting extensions of two distinct instruction-set architectures	712/210
132	US 5852727 A	☒	Instruction scanning unit for locating instructions via parallel scanning of start and end byte information	712/215

Return PC	Call Tag	Ret. Tag	CV	RV	CRV	IXC	IStart

Return Stack Storage
252A

. . . .

FIG. 2A

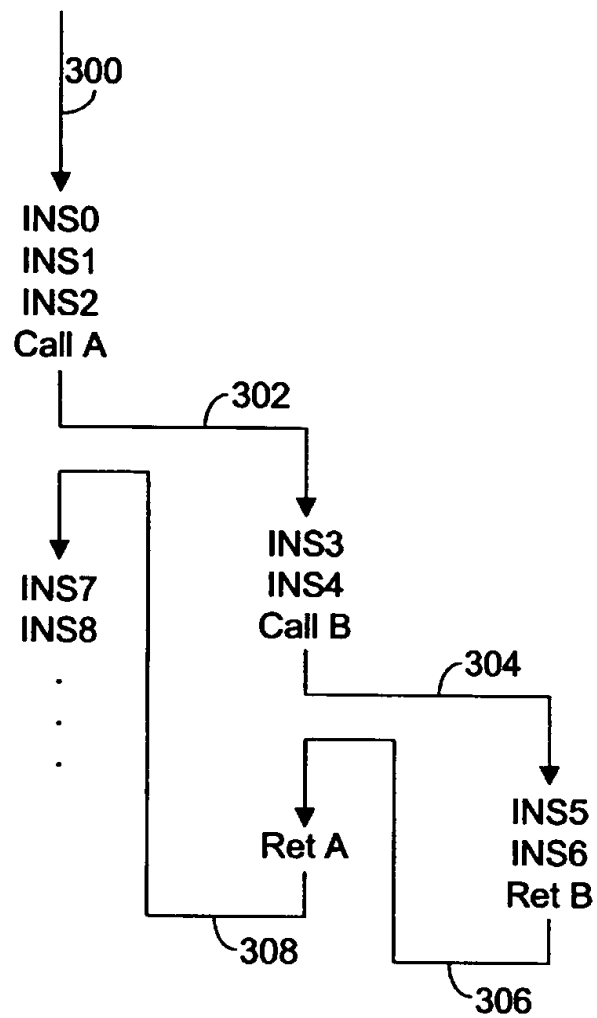|	Docum ent ID	U	Title	Current OR
133	US 58505 32 A	☒	Invalid instruction scan unit for detecting invalid predecode data corresponding to instructions being fetched	712/213
134	US 58484 33 A	☒	Way prediction unit and a method for operating the same	711/137
135	US 58482 69 A	☒	Branch predicting mechanism for enhancing accuracy in branch prediction by reference to data	712/239
136	US 58389 40 A	☒	Method and apparatus for rotating active instructions in a parallel data processor	712/216
137	US 58357 45 A	☒	Hardware instruction scheduler for short execution unit latencies	712/215
138	US 58322 97 A	☒	Superscalar microprocessor load/store unit employing a unified buffer and separate pointers for load and store operations	710/5
139	US 58322 49 A	☒	High performance superscalar alignment unit	712/204
140	US 58288 74 A	☒	Past-history filtered branch prediction	712/240
141	US 58260 71 A	☒	Parallel mask decoder and method for generating said mask	712/224
142	US 58225 75 A	☒	Branch prediction storage for storing branch prediction information such that a corresponding tag may be routed with the branch instruction	712/239
143	US 58225 74 A	☒	Functional unit with a pointer for mispredicted resolution, and a superscalar microprocessor employing the same	712/233
144	US 58225 59 A	☒	Apparatus and method for aligning variable byte-length instructions to a plurality of issue positions	712/214
145	US 58225 58 A	☒	Method and apparatus for predecoding variable byte-length instructions within a superscalar microprocessor	712/213
146	US 58190 59 A	☒	Predecode unit adapted for variable byte-length instruction set processors and method of operating the same	712/213
147	US 58190 57 A	☒	Superscalar microprocessor including an instruction alignment unit with limited dispatch to decode units	712/204
148	US 58157 00 A	☒	Branch prediction table having pointers identifying other branches within common instruction cache lines	712/240
149	US 58128 38 A	☒	Branch history table	712/239
150	US 58092 71 A	☒	Method and apparatus for changing flow of control in a processor	712/208
151	US 57940 63 A	☒	Instruction decoder including emulation using indirect specifiers	712/23
152	US 57940 28 A	☒	Shared branch prediction structure	712/240
153	US 57939 40 A	☒	Data processing apparatus	714/1
154	US 57817 89 A	☒	Superscaler microprocessor employing a parallel mask decoder	712/23
155	US 57747 10 A	☒	Cache line branch prediction scheme that shares among sets of a set associative cache	712/238

300

INS0
INS1
INS2
Call A

302

INS7
INS8
.
.
.

INS3
INS4
Call B

304

Ret A

INS5
INS6
Ret B

308

306

FIG. 3

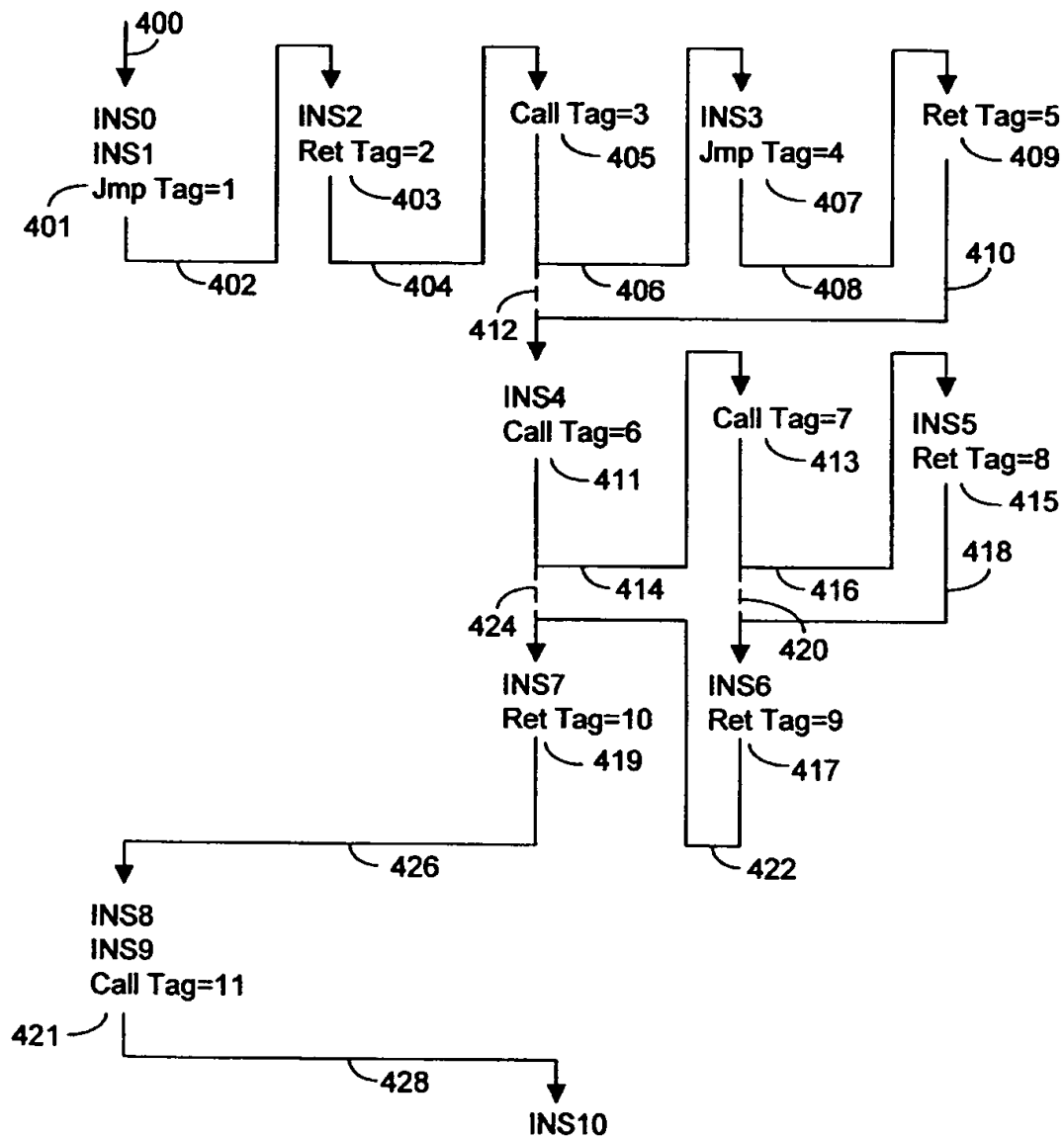|	Document ID	U	Title	Current OR
156	US 5768610 A	☒	Lookahead register value generator and a superscalar microprocessor employing same	712/23
157	US 5764946 A	☒	Superscalar microprocessor employing a way prediction unit to predict the way of an instruction fetch address and to concurrently provide a branch prediction address corresponding to the fetch address	712/239
158	US 5758143 A	☒	Method for updating a branch history table in a processor which resolves multiple branches in a single cycle	712/240
159	US 5758112 A	☒	Pipeline processor with enhanced method and apparatus for restoring register-renaming information in the event of a branch misprediction	712/217
160	US 5752259 A	☒	Instruction cache configured to provide instructions to a microprocessor having a clock cycle time less than a cache access time of said instruction cache	711/125
161	US 5752069 A	☒	Superscalar microprocessor employing away prediction structure	712/23
162	US 5748976 A	☒	Mechanism for maintaining data coherency in a branch history instruction cache	712/240
163	US 5740418 A	☒	Pipelined processor carrying out branch prediction by BTB	712/239
164	US 5740415 A	☒	Instruction supplying apparatus with a branch target buffer having the contents so updated as to enhance branch prediction accuracy	712/238
165	US 5737590 A	☒	Branch prediction system using limited branch target buffer updates	712/238
166	US 5734881 A	☒	Detecting short branches in a prefetch buffer using target location information in a branch target cache	712/238
167	US 5729728 A	☒	Method and apparatus for predicting, clearing and redirecting unpredicted changes in instruction flow in a microprocessor	712/234
168	US 5721855 A	☒	Method for pipeline processing of instructions by controlling access to a reorder buffer using a register file outside the reorder buffer	712/218
169	US 5715440 A	☒	Branch instruction executing device for tracing branch instruments based on instruction type	712/233
170	US 5704054 A	☒	Counterflow pipeline processor architecture for semi-custom application specific IC's	712/212
171	US 5701448 A	☒	Detecting segment limit violations for branch target when the branch unit does not supply the linear address	712/233
172	US 5696919 A	☒	Accessing a desk-type recording medium having reproduction control data	345/841
173	US 5687338 A	☒	Method and apparatus for maintaining a macro instruction for refetching in a pipelined processor	712/205
174	US 5687110 A	☒	Array having an update circuit for updating a storage location with a value stored in another storage location	365/154
175	US 5644745 A	☒	Apparatus for replacing data availability information for an instruction subsequent to a branch with previous availability information upon branch prediction failure	712/216
176	US 5642500 A	☒	Method and apparatus for controlling instruction in pipeline processor	712/233
177	US 5634119 A	☒	Computer processing unit employing a separate millicode branch history table	712/240

400

INS0
INS1
Jmp Tag=1
401
402

INS2
Ret Tag=2
403
404

Call Tag=3
405
406
412

INS3
Jmp Tag=4
407
408

Ret Tag=5
409
410

INS4
Call Tag=6
411
414
424

Call Tag=7
413
416
420

INS5
Ret Tag=8
415
418

INS7
Ret Tag=10
419
426

INS6
Ret Tag=9
417
422

INS8
INS9
Call Tag=11
421
428

INS10

FIG. 4A

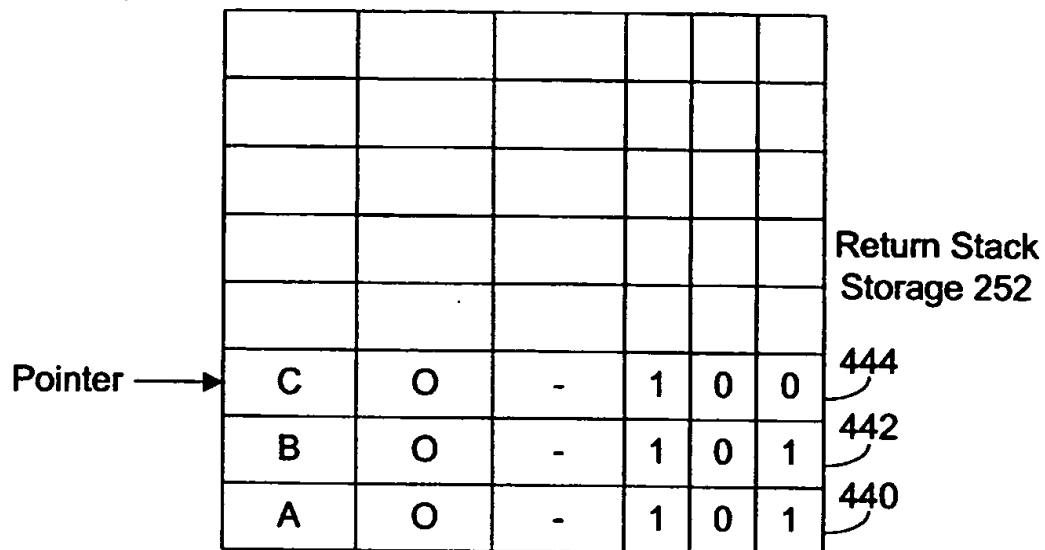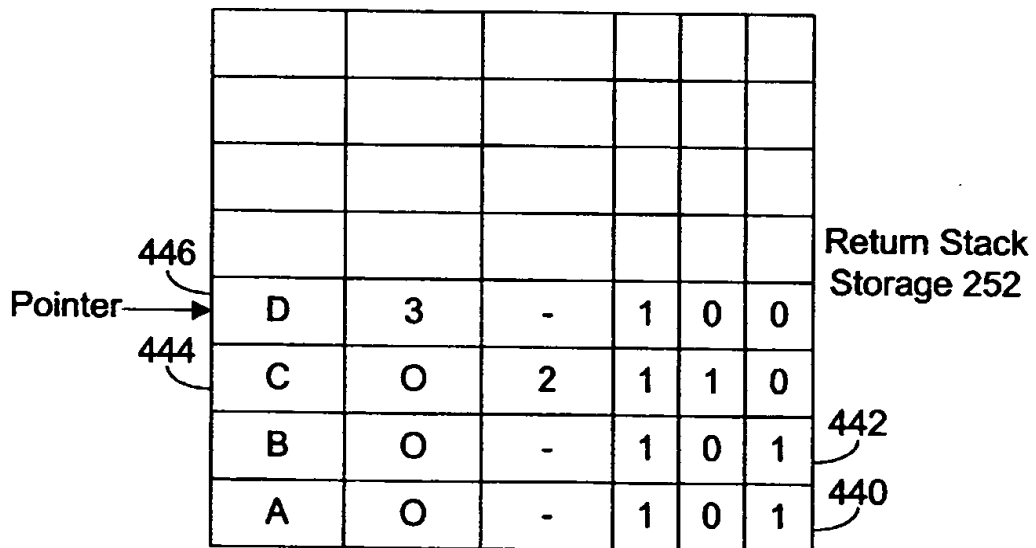|	Docum ent ID	U	Title	Current OR
178	US 56320 28 A	☒	Hardware support for fast software emulation of unimplemented instructions	703/26
179	US 56196 62 A	☒	Memory reference tagging	712/216
180	US 55840 09 A	☒	System and method of retiring store data from a write buffer	711/117
181	US 55840 01 A	☒	Branch target buffer for dynamically predicting branch instruction outcomes using a predicted branch history	712/238
182	US 55817 19 A	☒	Multiple block line prediction	712/207
183	US 55772 17 A	☒	Method and apparatus for a branch target buffer with shared branch pattern tables for associated branch predictions	712/200
184	US 55641 18 A	☒	Past-history filtered branch prediction	712/240
185	US 55532 55 A	☒	Data processor with programmable levels of speculative instruction fetching and method of operation	712/235
186	US 55220 53 A	☒	Branch target and next instruction address calculation in a pipeline processor	711/213
187	US 55198 41 A	☒	Multi instruction register mapper	711/202
188	US 55176 14 A	☒	Data compression/encryption processing apparatus	714/1
189	US 55111 75 A	☒	Method an apparatus for store-into-instruction-stream detection and maintaining branch prediction cache consistency	712/216
190	US 55048 70 A	☒	Branch prediction device enabling simultaneous access to a content-addressed memory for retrieval and registration	712/238
191	US 54715 97 A	☒	System and method for executing branch instructions wherein branch target addresses are dynamically selectable under programmer control from writable branch address tables	711/215
192	US 54540 89 A	☒	Branch look ahead adder for use in an instruction pipeline sequencer with multiple instruction decoding	711/213
193	US 54248 82 A	☒	Signal processor for discriminating recording data	360/46
194	US 53945 30 A	☒	Arrangement for predicting a branch target address in the second iteration of a short loop	712/240
195	US 53677 03 A	☒	Method and system for enhanced branch history prediction accuracy in a superscalar processor system	712/23
196	US 53275 36 A	☒	Microprocessor having branch prediction function	712/238
197	US 53136 34 A	☒	Computer system branch prediction of subroutine returns	712/240
198	US 52281 31 A	☒	Data processor with selectively enabled and disabled branch prediction operation	712/240
199	US 52261 30 A	☒	Method and apparatus for store-into-instruction-stream detection and maintaining branch prediction cache consistency	712/238
200	US 51931 56 A	☒	Data processor with pipeline which disables exception processing for non-taken branches	712/239

C	O	-	1	0	0	444
B	O	-	1	0	1	442
A	O	-	1	0	1	440

Return Stack Storage 252

Pointer →

FIG. 4B

D	3	-	1	0	0
C	O	2	1	1	0
B	O	-	1	0	1	442
A	O	-	1	0	1	440

Return Stack Storage 252

446

Pointer →

444

FIG. 4C

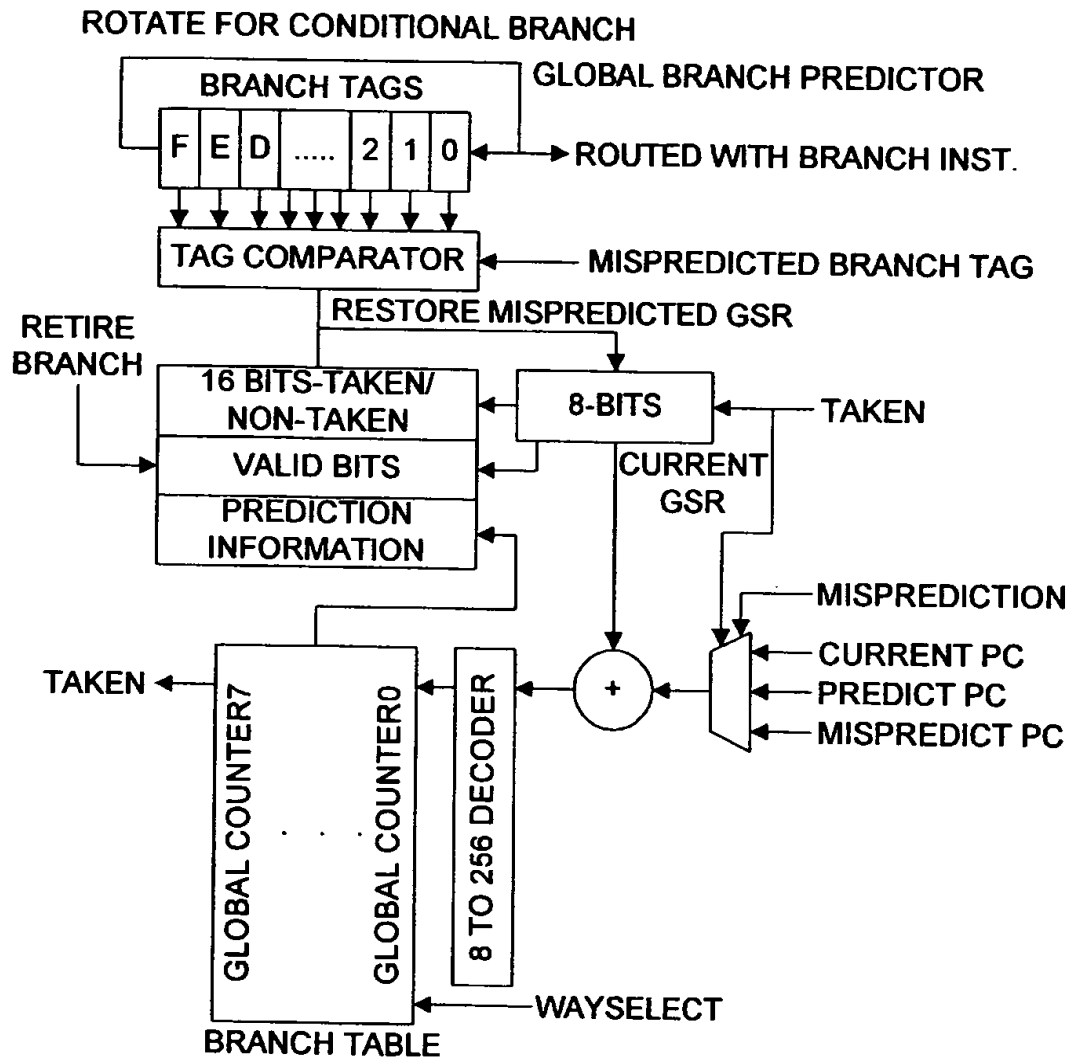|	L #	Hits	Search Text	DBs
1	L2	869	(access$3 writ$3 updat$3) near10 ((branch near5 (table history predict$3)) bht)	USPAT; US-PGPUB
2	L4	207	(access$3 writ$3 updat$3) near10 ((branch near5 (table history predict$3)) bht)	EPO; JPO; DERWENT; IBM_TDB
3	L7	220	2 near10 (control controlled controlling simultaneous$3 clock cycle)	USPAT; US-PGPUB
4	L6	29	4 near10 (control controlled controlling simultaneous$3 clock cycle)	EPO; JPO; DERWENT; IBM_TDB
5	L8	1245	(access$3 writ$3 updat$3) near20 ((branch near20 (table history predict$3)) bht)	USPAT; US-PGPUB
6	L10	258	(access$3 writ$3 updat$3) near20 ((branch near20 (table history predict$3)) bht)	EPO; JPO; DERWENT; IBM_TDB
7	L12	11	10 near30 (control controlled controlling simultaneous$3 clock cycle) not 6	EPO; JPO; DERWENT; IBM_TDB
8	L9	73	8 near30 (control controlled controlling simultaneous$3 clock cycle) not 7	USPAT; US-PGPUB

ROTATE FOR CONDITIONAL BRANCH

BRANCH TAGS	GLOBAL BRANCH PREDICTOR

| F | E | D | ..... | 2 | 1 | 0 | ◄──► ROUTED WITH BRANCH INST. |

TAG COMPARATOR ◄─── MISPREDICTED BRANCH TAG

RETIRE BRANCH

RESTORE MISPREDICTED GSR

| 16 BITS-TAKEN/NON-TAKEN |
| VALID BITS |
| PREDICTION INFORMATION |

8-BITS ◄─── TAKEN

CURRENT GSR

MISPREDICTION
CURRENT PC
PREDICT PC
MISPREDICT PC

TAKEN ◄───

GLOBAL COUNTER7 · · · GLOBAL COUNTER0

8 TO 256 DECODER

+

WAYSELECT

BRANCH TABLE

FIG. 14

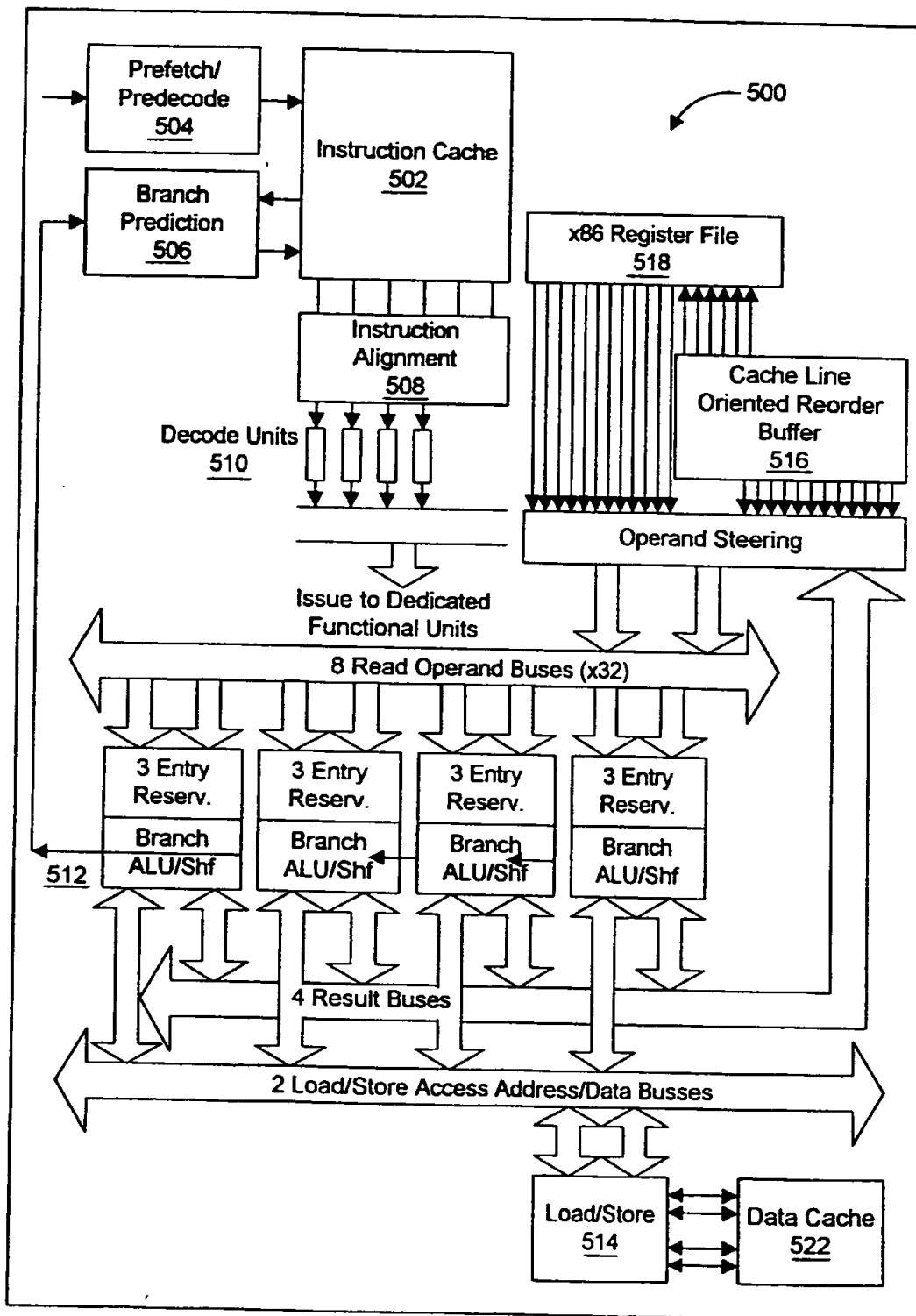|	Docum ent ID	U	Title	Current OR
1	JP 08286 913 A	☐	METHOD AND DEVICE FOR GENERATING ADDRESS OF INSTRUCTION TO BE FINISHED NEXT IN PIPELINE PROCESSOR	
2	JP 04148 359 A	☒	HIGH SPEED FRONT END PROCESSOR	
3	JP 01284 927 A	☒	INFORMATION PROCESSOR	
4	JP 63208 941 A	☒	MANAGING SYSTEM FOR UPDATING HISTORY OF PROGRAM PATH	
5	JP 61257 041 A	☒	PACKET EXCHANGE SYSTEM OF HIERARCHY EXCHANGE SYSTEM	
6	JP 55041 515 A	☒	HOME COMPUTER	
7	EP 73683 0 A1	☒	Method and apparatus for reconstructing the address of the next instruction to be completed in a pipelined processor	
8	NN920 1330	☒	Using History to Improve the Handling of Address Generation Interlocks in Branch Instructions.	
9	US 20020 07833 2 A	☒	Computer system includes bank control logic to ensure that no two look-ups access same bank in same clock cycle	
10	GB 23213 23 A	☐	Branch prediction method for computer program - involves accessing branch target buffer using pointer of instruction whose interval from fetch of branch instruction is smallest	
11	EP 21074 2 A	☒	Processing information through multi-program system - using with computer program sequence, command shell with set of programs and program control files	

Prefetch/
Predecode
504

Instruction Cache
502

— 500

Branch
Prediction
506

x86 Register File
518

Instruction
Alignment
508

Cache Line
Oriented Reorder
Buffer
516

Decode Units
510

Operand Steering

Issue to Dedicated
Functional Units

8 Read Operand Buses (x32)

3 Entry Reserv.	3 Entry Reserv.	3 Entry Reserv.	3 Entry Reserv.
Branch ALU/Shf	Branch ALU/Shf	Branch ALU/Shf	Branch ALU/Shf

512

4 Result Buses

2 Load/Store Access Address/Data Busses

Load/Store
514

Data Cache
522

FIG. 5

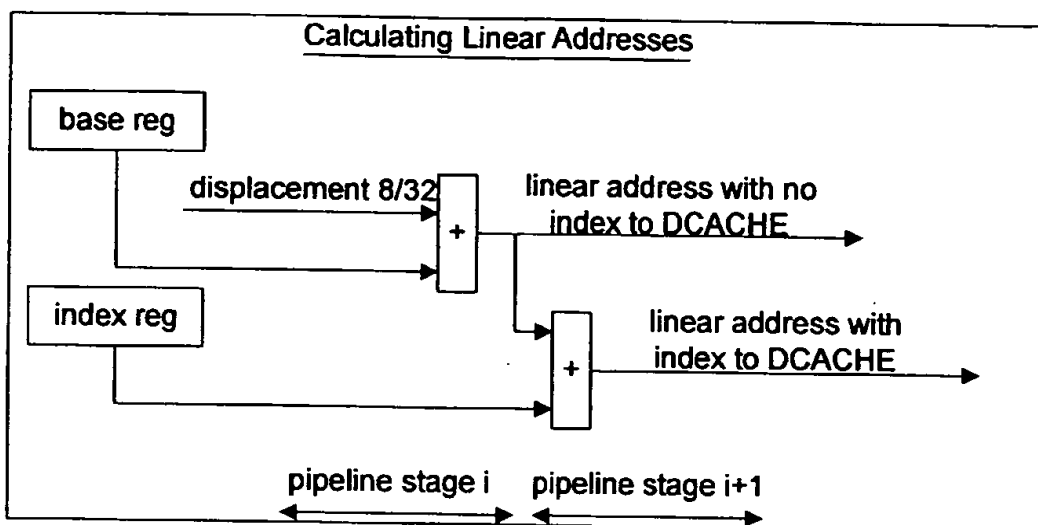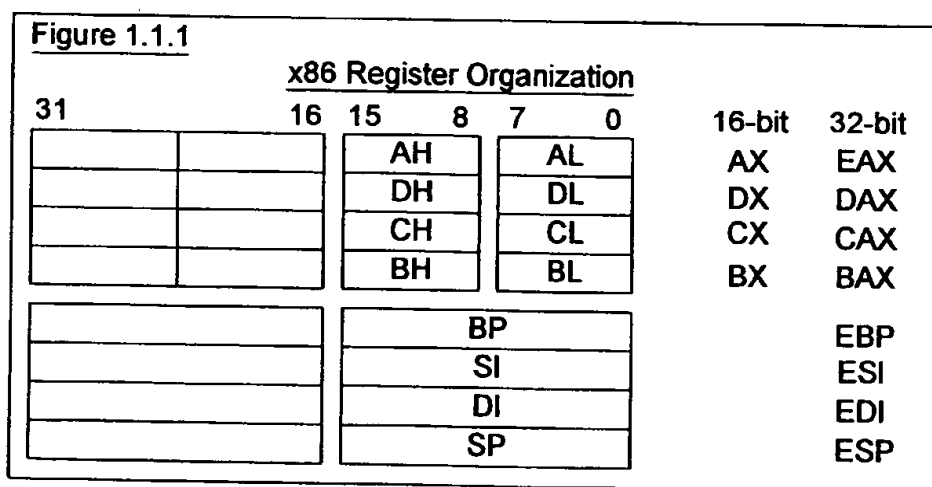|	Docum ent ID	U	Title	Current OR
1	US 20020 13992 7 A1	☐	Method of and system for detecting and correcting mode switching in diffractive-based laser scanning·systems	250/235
2	US 20020 11213 1 A1	☒	Specifying access control and caching on operands	711/141
3	US 20020 06935 1 A1	☒	Memory data access structure and method suitable for use in a processor	712/239
4	US 20020 06908 6 A1	☒	Web linked database for tracking clinical activities and competencies and evaluation of program resources and program outcomes	705/2
5	US 20020 04139 6 A1	☒	Storage medium control apparatus, image forming apparatus using the same, and control method therefor	358/1.1 7
6	US 20020 03567 7 A1	☒	METHOD AND APPARATUS FOR PRE-PROCESSING INSTRUCTIONS FOR A PROCESSOR	712/217
7	US 20010 03749 7 A1	☒	Apparatus and method for generating optimization objects	717/160
8	US 64424 79 B1	☒	Method and apparatus for a location sensitive database	701/213
9	US 64381 81 B1	☒	Efficient metric memory configuration for a Viterbi decoder	375/341
10	US 64250 76 B1	☒	Instruction prediction based on filtering	712/239
11	US 63603 18 B1	☒	Configurable branch prediction for a processor performing speculative execution	712/240
12	US 63518 39 B1	☒	State metric memory of viterbi decoder and its decoding method	714/795
13	US 63393 83 B1	☒	Traffic signal control apparatus optimizing signal control parameter by rolling horizon scheme	340/907
14	US 62928 79 B1	☒	Method and apparatus to specify access control list and cache enabling and cache coherency requirement enabling on individual operands of an instruction of a computer	711/214
15	US 62826 39 B1	☒	Configurable branch prediction for a processor performing speculative execution	712/240
16	US 62821 83 B1	☒	Method for authorizing couplings between devices in a capability addressable network	370/338
17	US 62302 60 B1	☒	Circuit arrangement and method of speculative instruction execution utilizing instruction history caching	712/239
18	US 61700 54 B1	☒	Method and apparatus for predicting target addresses for return from subroutine instructions utilizing a return address cache	712/242
19	US 61548 33 A	☒	System for recovering from a concurrent branch target buffer read with a write allocation by invalidating and then reinstating the instruction pointer	712/238
20	US 61449 82 A	☒	Pipeline processor and computing system including an apparatus for tracking pipeline resources	709/104

Calculating Linear Addresses

base reg

displacement 8/32   linear address with no index to DCACHE

index reg

+   linear address with index to DCACHE

pipeline stage i   pipeline stage i+1

FIG. 6

Figure 1.1.1

x86 Register Organization

31	16	15	8	7	0	16-bit	32-bit
		AH		AL		AX	EAX
		DH		DL		DX	DAX
		CH		CL		CX	CAX
		BH		BL		BX	BAX
		BP					EBP
		SI					ESI
		DI					EDI
		SP					ESP

FIG. 7

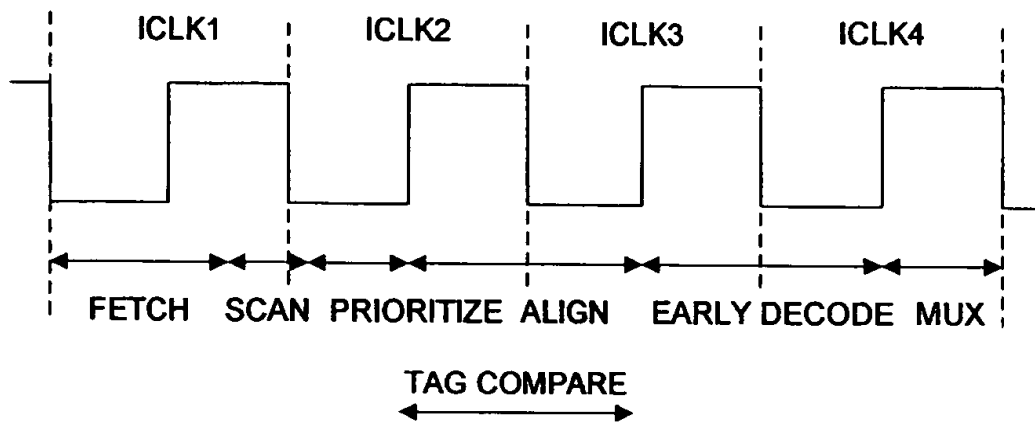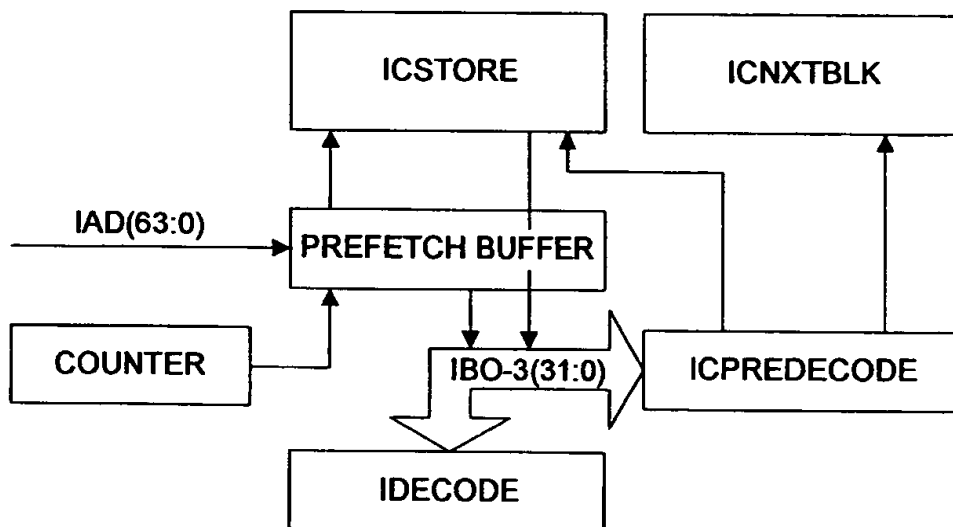|	Docum ent ID	U	Title	Current OR
21	US 61087 77 A	☒	Configurable branch prediction for a processor performing speculative execution	712/240
22	US 61087 75 A	☒	Dynamically loadable pattern history tables in a multi-task microprocessor	712/240
23	US 61087 69 A	☒	Dependency table for reducing dependency checking hardware	712/216
24	US 61015 86 A	☒	Memory access control circuit	711/163
25	US 60921 87 A	☒	Instruction prediction based on filtering	712/239
26	US 60852 38 A	☒	Virtual LAN system	709/223
27	US 60524 41 A	☒	Voice response service apparatus	379/88. 16
28	US 60444 59 A	☒	Branch prediction apparatus having branch target buffer for effectively processing branch instruction	712/237
29	US 59408 57 A	☒	Instruction cache memory apparatus with advanced read function that predicts whether to read out a next instruction block including an address register, a counter and a selector	711/137
30	US 59403 78 A	☒	Call control in exchange suitable for intelligent network	370/259
31	US 58600 18 A	☒	Method for tracking pipeline resources in a superscalar processor	712/23
32	US 58570 89 A	☒	Floating point stack and exchange instruction	712/222
33	US 58451 02 A	☒	Determining microcode entry points and prefix bytes using a parallel logic technique	712/211
34	US 58394 83 A	☒	Beverage dispenser with serving time monitor	141/1
35	US 58357 54 A	☒	Branch prediction system for superscalar processor	712/239
36	US 58156 99 A	☒	Configurable branch prediction for a processor performing speculative execution	712/239
37	US 58092 94 A	☒	Parallel processing unit which processes branch instructions without decreased performance when a branch is taken	712/233
38	US 58058 78 A	☒	Method and apparatus for generating branch predictions for multiple branch instructions indexed by a single instruction pointer	712/239
39	US 57969 88 A	☒	Method and system using dedicated location to share information between real mode and protected mode drivers	709/321
40	US 57963 56 A	☒	Data compressing apparatus, data restoring apparatus and data compressing/restoring system	341/51
41	US RE357 94 E	☒	System for reducing delay for execution subsequent to correctly predicted branch instruction using fetch information stored with each block of instructions in cache	712/239
42	US 56969 55 A	☒	Floating point stack and exchange instruction	712/222
43	US 56842 60 A	☒	Apparatus and method for generation and synthesis of audio	84/604

ICLK1	ICLK2	ICLK3	ICLK4

FETCH   SCAN   PRIORITIZE   ALIGN   EARLY DECODE   MUX

TAG COMPARE

FIG. 8

ICSTORE

ICNXTBLK

IAD(63:0)

PREFETCH BUFFER

COUNTER

IBO-3(31:0)

ICPREDECODE

IDECODE

FIG. 9

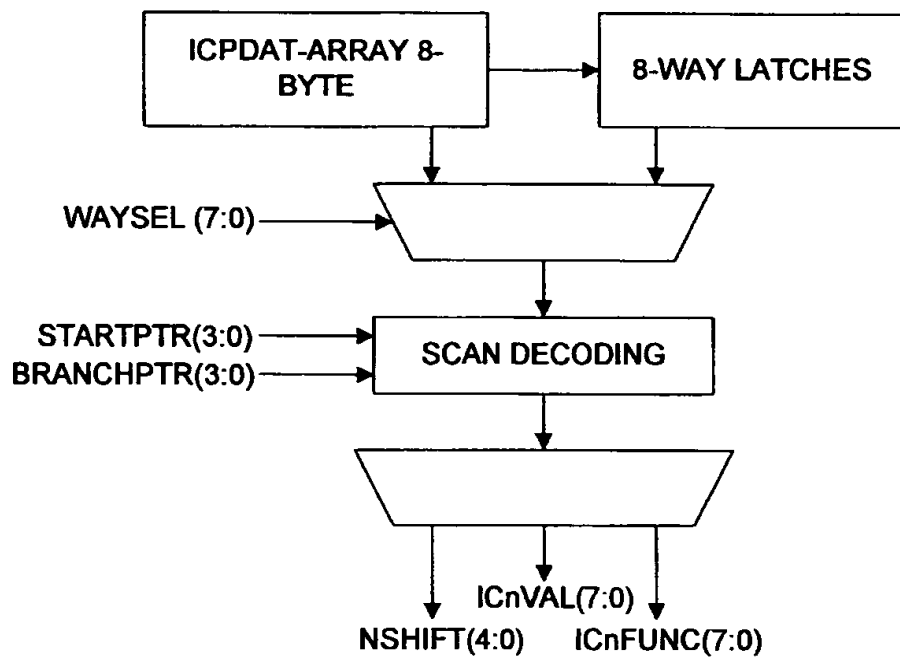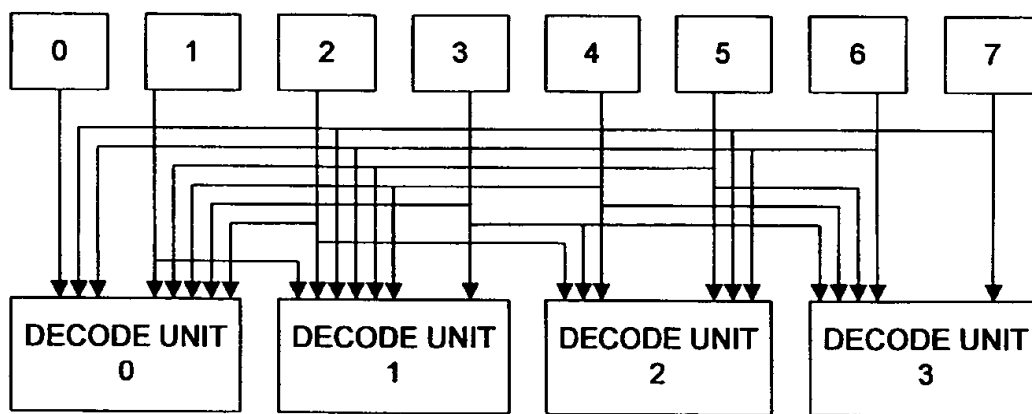|	Document ID	U	Title	Current OR
44	US 5675645 A	☒	Method and apparatus for securing executable programs against copying	713/187
45	US 5655096 A	☒	Method and apparatus for dynamic scheduling of instructions to ensure sequentially coherent data in a processor employing out-of-order execution	712/200
46	US 5649108 A	☒	Combined progressive and source routing control for connection-oriented communications networks	709/241
47	US 5634027 A	☒	Cache memory system for multiple processors with collectively arranged cache tag memories	711/3
48	US 5604887 A	☒	Method and system using dedicated location to share information between real and protected mode device drivers	703/27
49	US 5592634 A	☒	Zero-cycle multi-state branch cache prediction data processing system and method thereof	712/239
50	US 5568631 A	☒	Multiprocessor system with a shared control store accessed with predicted addresses	712/248
51	US 5481364 A	☒	Apparatus for adaptively generating a decoder table for variable-length codes using a stored coding table	382/244
52	US 5463746 A	☒	Data processing system having prediction by using an embedded guess bit of remapped and compressed opcodes	712/240
53	US 5459845 A	☒	Instruction pipeline sequencer in which state information of an instruction travels through pipe stages until the instruction execution is completed	712/248
54	US 5454117 A	☒	Configurable branch prediction for a processor performing speculative execution	712/23
55	US 5453927 A	☒	Data processor for processing branch instructions	712/235
56	US 5388239 A	☒	Operand address modification system	711/220
57	US 5353421 A	☒	Multi-prediction branch prediction mechanism	712/240
58	US 5345571 A	☒	System for controlling branch history table	712/240
59	US 5317700 A	☒	Program history for pipelined processor including temporary storage queues for storing branch addresses	712/240
60	US 5305375 A	☒	Information service apparatus	379/88.27
61	US 5305217 A	☒	Method and system for controlling automatic guided vehicle	701/25
62	US 5303234 A	☒	Random access data communication system with slot assignment capability for contending users	370/442
63	US 5295264 A	☒	Modularly structured integrated services digital network (ISDN) communication system	709/103
64	US 5267190 A	☒	Simultaneous search-write content addressable memory	365/49
65	US 5222064 A	☒	Bridge apparatus	370/401
66	US 5203006 A	☒	System for selecting next instruction address between unit incremented address and address from table specified by operating condition signals	711/213

FIG. 10

FIG. 11

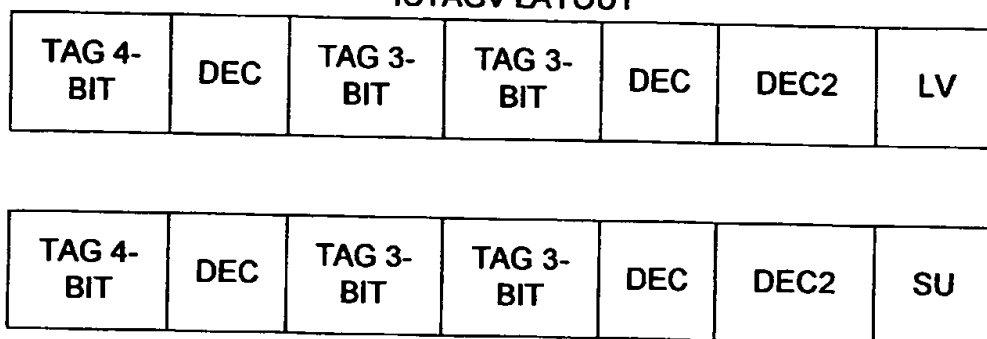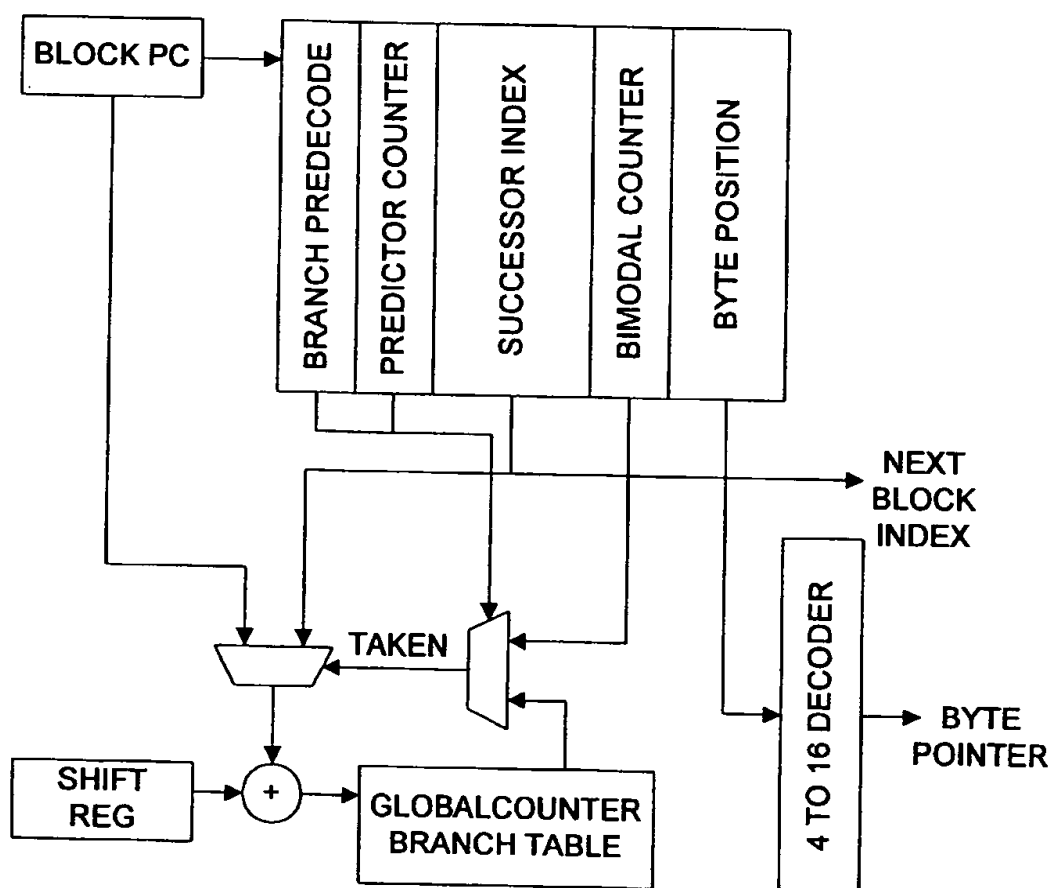|	Docum ent ID	U	Title	Current OR
67	US 51366 96 A	☒	High-performance pipelined central processor for predicting the occurrence of executing single-cycle instructions and multicycle instructions	712/240
68	US 50723 64 A	☒	Method and apparatus for recovering from an incorrect branch prediction in a processor that executes a family of instructions in parallel	712/215
69	US 47605 19 A	☒	Data processing apparatus and method employing collision detection and prediction	712/217
70	US 47501 12 A	☒	Data processing apparatus and method employing instruction pipelining	712/217
71	US 45478 46 A	☒	Accessory interface circuit for universal multi-station document inserter	700/9
72	US 44505 25 A	☒	Control unit for a functional processor	712/243
73	US 43631 03 A	☒	Device for following and estimating the local state of picture contours	382/242

ICTAGV LAYOUT

TAG 4-BIT	DEC	TAG 3-BIT	TAG 3-BIT	DEC	DEC2	LV

TAG 4-BIT	DEC	TAG 3-BIT	TAG 3-BIT	DEC	DEC2	SU

FIG. 12

ICNXTBLK

FIG. 13